

# Project “egov\_INNO”

“E-government services and tools from regional governments and regional development bodies to support and coordinate the regional research and innovation capital”

## BDI System Architecture

Author: InnovaPuglia



## ***1. Introduction***

BDI (Banca Dati Innovazione) is an online data warehouse that allows to support the user (enterprise, investor or public administration) in querying a database built from heterogeneous regional databases realised with very vertical objectives. The aim is to capitalize investments in research, innovation and material activities, provided on the regional territory.

It is made using low-cost open source technologies.

## ***2. System architecture and functionalities***

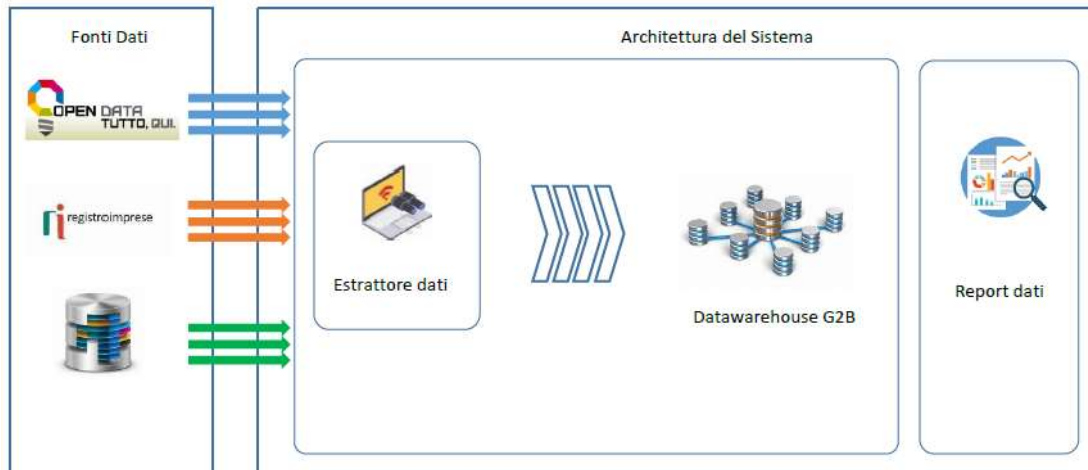
### **2.1. General overview**

BDI feeds information from different and heterogeneous sources.

The following sources of information have been identified:

- Open Data system reachable at the address [data.puglia.it](http://data.puglia.it);
- Chamber of Commerce data for all enterprises, including innovative enterprises;
- some "vertical/management" databases, whose information is made available by the respective managers through web service.

It is also being acquired Open Patent Services (or OPS), that provides web services for machine-to-machine queries that deliver production stable patent data from the European Patent Office (EPO). This will make it possible to obtain, among the research results, information on patents registered by companies in Contracting States to the European Patent Convention.



The logical model described in the figure includes the following macro-components:

- **Data Sources:** are the sources of information from which the system will draw the data necessary for its purpose.
- **Data Extractor:** is the core component of the System that deals with:
  - "extract" data from information sources and in accordance with the modalities and timing defined in the System;
  - "transform" them into tables and upload them to the database of the system on which the searches are subsequently built.
- **Datawarehouse:** is the database that collects all the extracted data making them readable to the user for the purposes of the project
- **Data Report:** is the system component that offers the end user the interface to explore the datawarehouse database.

## 2.2. Database Design

The system database consists of a series of tables whose structure has been created in accordance with the structure of data acquired from external sources.

In addition to these tables, the database contains the "guide" tables around which the final services to be provided revolve, such as geographical area table, sector table etc.

In addition, there will be some tables of "system" necessary for the functioning of the System and related to the organization of users: roles, users, user profiles.

Below is an image of the database structure:



## 2.4. Roles and Permissions

Below is the classification of users who interact with the system and their main purposes.

User Code	Description	Finality
ROLE_RESPONSABILE	Apulia Region User Project User	<ul style="list-style-type: none"> <li>guided research carried out</li> <li>free searches carried out</li> <li>access to data source display</li> <li>access database structure</li> </ul>
ROLE_AMMINISTRATORE_RP	Administrator	<ul style="list-style-type: none"> <li>managing registration and user profiling</li> <li>guided research carried out</li> <li>free searches carried out</li> <li>access to data source display</li> <li>access database structure</li> </ul>
ROLE_OPERATORE	Generic User	<ul style="list-style-type: none"> <li>managing registration and user profiling</li> <li>guided research carried out (all sources of data, except that of the Chamber of Commerce)</li> </ul>

## 3. Software Description

### 3.1. Software Elements

The software architecture of the system is subdivided into the following levels:

- The **User Interface (UI)** layer consists of HTML 5 page structure, based on the Angularjs framework.
- The **Application layer** consists of the Java classes run within the Tomcat application server that represent the web application controllers.
- The **Domain** layer where there are Java classes that contain the application logic used by all sections of the application. This layer includes the following groups of packages:
  - Entity repositories - Spring repositories useful for handling entities managed by the application

- Entities - JPA entities mapping database tables
- External service adapters - Adapters used by controllers for interfacing with external systems
- The layer **Technical services** in which the lowest level components are present divided into the following groups of packages:
  - Mysql Data Access Objects - Classes managed by JPA for accessing the Mysql database.
  - External service clients - Client for access to external services via web service or REST service.

## ***4. Technological components***

The following are the technological components used by the application:

- Centos operating system
- Mysql database
- Frontend: HTML5 and Angularjs
- Backend: Web application developed in Java and executed within the Apache Tomcat application server